Αυτόματη Κατασκευή Ωρολογίων Προγραμμάτων μέσω Προγραμματισμού με Περιορισμούς

Νικόλαος Ποθητός*

Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Πανεπιστημιούπολη, 157 84 Αθήνα
pothitos@di.uoa.gr

Περίληψη Η αυτόματη κατασκευή ωρολογίων προγραμμάτων είναι ένα γνωστό πρόβλημα χρονοπρογραμματισμού που αντιμετωπίζεται συνήθως με τεχνικές βελτιστοποίησης από τον χώρο της της Τεχνητής Νοημοσύνης και της Επιχειρησιακής Έρευνας. Στην παρούσα εργασία, χρησιμοποιήθηκε ο Προγραμματισμός με Περιορισμούς, μία σύγχρονη μεθοδολογία στην οποία η διατύπωση ενός προβλήματος γίνεται ανεξάρτητα από τον τρόπο επίλυσής του. Η κατάρτιση ωρολογίων προγραμμάτων για πανεπιστήμια και σχολεία διατυπώθηκε σαν ένα απλό Πρόβλημα Ικανοποίησης Περιορισμών (ΠΙΠ). Επιπρόσθετα, κατασκευάστηκε και χρησιμοποιήθηκε ένας επιλυτής γενικών ΠΙΠ, για την αναζήτηση λύσεων στο πρόβλημα.

Λέξεις-κλειδιά: προγραμματισμός με περιορισμούς, χρονοπρογραμματισμός, ωρολόγιο πρόγραμμα

1 Εισαγωγή

Ο χρονοπρογραμματισμός δραστηριοτήτων που εξαρτώνται από πόρους, αποτελεί εδώ και δεκαετίες το επίκεντρο σημαντικής έρευνας [13,8]. Έτσι, έχουν αναπτυχθεί πολλές τεχνικές για την αντιμετώπιση τέτοιου είδους προβλημάτων, ένα από τα οποία είναι και η κατάρτιση ωρολογίων προγραμμάτων (timetabling).

Κάθε εκπαιδευτικό ίδρυμα λοιπόν, έχει να κατασκευάσει το δικό του ωρολόγιο πρόγραμμα· έχουν γίνει γνωστοί πολλοί τρόποι για να γίνει η κατασκευή αυτή αυτόματα [4]. Οι ομοιότητες του προβλήματος με αυτό του χρωματισμού γράφων —αν δούμε τα μαθήματα σαν κόμβους και τους περιορισμούς που τα συνδέουν σαν ακμές— χρησιμοποιήθηκαν για την επινόηση κοινών διαδικασιών επίλυσής τους [6]. Ακόμα, η κατάρτιση ωρολογίων προγραμμάτων έχει συσχετισθεί και με τη γενική κατηγορία των προβλημάτων δικτύων ροής (network flow problems) [14]. Μία άλλη μέθοδος περιλαμβάνει το «κομμάτιασμα» του προβλήματος σε υπο-προβλήματα (clustering) [20], χωρίς όμως να έχει δώσει πολύ καλά αποτελέσματα. Πιο καλά αποτελέσματα έχουν δώσει τελευταία η εφαρμογή της λογικής βασισμένης σε περιπτώσεις (case-based reasoning) στο πρόβλημα που εξετάζουμε [3].

* Επιβλέπων Καθηγητής: Παναγιώτης Σταματόπουλος

Λόγω της δυσκολίας εύρεσης καλών λύσεων στο πρόβλημα (δηλαδή λύσεων που θα ικανοποιούν π.χ. τους φοιτητές και τους καθηγητές ενός ιδρύματος), έχουν εφαρμοστεί πάνω σε αυτό πολλές μεταευριστικές μέθοδοι (metaheuristics). Γενικά, αυτές οι μέθοδοι παίρνουν μία αρχική κατάσταση των μεταβλητών του προβλήματος και προσπαθούν επαναληπτικά να μεταβούν σε μία άλλη κατάσταση που βρίσκεται πιο κοντά σε μία λύση (η οποία θα είναι καλύτερη από την τρέχουσα λύση που έχει ήδη βρεθεί, εφόσον υπάρχει). Το μειονέκτημα αυτών των μεθόδων είναι ότι μπορεί να «εγκλωβιστούν» σε *τοπικά* μέγιστα, αλλά παρ' όλα αυτά υπάρχουν τρόποι να τα ξεπερνάμε. Η προσομοιωμένη ανόπτηση (simulated annealing), οι γενετικοί αλγόριθμοι (genetic algorithms), η ταμπού-αναζήτηση (tabu-search) και άλλες ευρέως χρησιμοποιούμενες μέθοδοι στην περιοχή της Τεχνητής Νοημοσύνης είχαν εφαρμογή στο πρόβλημα το οποίο εξετάζουμε [7,19,5]. Συγκεκριμένα, οι μέθοδοι τοπικής αναζήτησης έχουν εξελιχθεί έτσι ώστε να παράγουν σχεδόν βέλτιστες λύσεις, αλλά είναι πιο χρονοβόρες από τις άμεσες μεθόδους [10]. Μία άλλη κοινή υλοποίηση εφαρμογών στην Τεχνητή Νοημοσύνη είναι η διατύπωση των περιορισμών του προβλήματος [2] σε ένα περιβάλλον Προγραμματισμού με Περιορισμούς, το οποίο εκμεταλλεύεται τη γλώσσα λογικού προγραμματισμού Prolog [9,18], ή άλλες προγραμματιστικές φιλοσοφίες (π.χ. αντικειμενοστραφείς [21]). Βέβαια, πάντοτε υπήρχαν έξυπνες και παραγωγικές ιδέες που συνδύαζαν δύο ή περισσότερες από τις παραπάνω τεχνικές [12]. Πάντως, πρόσφατες εργασίες στοχεύουν όλο και περισσότερο στον ορισμό κοινώς αποδεκτών κριτηρίων αναφορικά με την αποτελεσματικότητα των συστημάτων κατάρτισης ωρολογίων προγραμμάτων [17].

Ανέκαθεν, οι ερευνητικές εργασίες εστίαζαν στη μείωση του υπολογιστικού χρόνου για τις εφαρμογές κατασκευής ενός ωρολογίου προγράμματος. Σήμερα, παρ' όλα αυτά, το κύριο εμπόδιο για την εξάπλωση αυτών των εφαρμογών είναι η ανελαστικότητά τους· σε πολλές περιπτώσεις δεν μπορούν να προσαρμοστούν στις διαφορετικές απαιτήσεις/περιορισμούς που απαντώνται πηγαίνοντας από το ένα εκπαιδευτικό ίδρυμα στο άλλο. Η κύρια συνεισφορά λοιπόν του Προγραμματισμού με Περιορισμούς (Constraint Programming) είναι ο διαχωρισμός της διαδικασίας διατύπωσης ενός προβλήματος από τον μηχανισμό ο οποίος το επιλύει. Αυτό είναι το στοιχείο-κλειδί που έκανε δημοφιλή στους προγραμματιστές αυτή τη μεθοδολογία της Τεχνητής Νοημοσύνης. Σε αυτό το πλαίσιο κινηθήκαμε και στην παρούσα εργασία. Ορίσαμε ρητά τις οντότητες του προβλήματος και απλοποιήσαμε τους περιορισμούς που τις συνδέουν. (Αυτή η απλότητα κάνει το πρόβλημα μεταφέρσιμο στους περισσότερους επιλυτές που υποστηρίζουν Προγραμματισμό με Περιορισμούς.) Από εκεί και πέρα, η πρόσθεση ή η τροποποίηση περιορισμών είναι εύκολη υπόθεση και δεν επηρεάζει την υλοποίηση της αναζήτησης.

Σε αυτήν την εργασία,[1] αρχικά, το πρόβλημα μοντελοποιείται μαθηματικά. Έπειτα κάνουμε προσθήκες στην αρχική μοντελοποίηση έτσι ώστε να προκύψει ένα ΠΙΠ. Έχοντας ορίσει ένα ΠΙΠ —με τις μεταβλητές και τα κριτήρια βελτιστοποίησής του—, μία βελτιωμένη μέθοδος αναζήτησης περιορισμένων ασυμφωνιών (improved Limited Discrepancy Search) συνδυάζεται με άλλες μεθόδους όπως η τοπική αναζήτηση, καθώς και με εύστοχα ευριστικά, για την παραγωγή λύσεων. Όλα αυτά

[1] Στη διεύθυνση http://www.di.uoa.gr/˜grad0780/skoptiko υπάρχει διαθέσιμη η υλοποίηση της εργασίας.

υλοποιούνται μέσα από έναν επιλυτή γενικών ΠΙΠ που κατασκευάσθηκε και χρησιμοποιήθηκε στα πλαίσια αυτής της εργασίας.

2 Οντότητες, Ιδιότητες και Σχέσεις του Προβλήματος

Καταρχήν, ορίζουμε τις οντότητες του προβλήματος· οι περιγραφές τους θα πρέπει να είναι όσο πιο γενικές γίνεται [1]. Έστω D ο αριθμός των ημερών ενός ωρολογίου προγράμματος· κάθε ημέρα διαρκεί H διδακτικές περιόδους.

$$P = \{0, 1, \ldots, D \cdot H - 1\}.$$

Το P είναι ένα σύνολο αποτελούμενο από όλες τις διδακτικές περιόδους. (Η πραγματική διάρκεια κάθε διδακτικής περιόδου μπορεί να είναι αυθαίρετη — π.χ. 45 λεπτά. Ωστόσο θεωρούμε ότι μία διδακτική περίοδος δεν διαιρείται σε υπο-περιόδους.) Έστω $C \neq \emptyset$ το σύνολο των αιθουσών,[2] $T \neq \emptyset$ το σύνολο των δασκάλων/καθηγητών, $S \neq \emptyset$ το σύνολο των μαθημάτων (το «S» προέρχεται από τη λέξη «Subjects»), L το σύνολο των διαλέξεων και G το σύνολο των ομάδων μαθημάτων, με $g \subseteq S$, $\forall g \in G$.

$$\begin{aligned} C &= \{c_1,\ c_2,\ \ldots,\ c_{|C|}\}, \\ T &= \{t_1,\ t_2,\ \ldots,\ t_{|T|}\}, \\ S &= \{s_1,\ s_2,\ \ldots,\ s_{|S|}\}, \\ L &= \{\ell_1,\ \ell_2,\ \ldots,\ \ell_{|L|}\}, \\ G &= \{g_1,\ g_2,\ \ldots,\ g_{|G|}\}. \end{aligned}$$

Έτσι, κάθε ομάδα $g \in G$ αποτελείται από μαθήματα και κάθε μάθημα $s \in S$ αποτελείται από μία ή περισσότερες διαλέξεις $\ell \in L$. Αυτή η ιεραρχία χρησιμοποιείται για την περιγραφή δύο συνθηκών που παρατηρούνται στα περισσότερα ακαδημαϊκά ιδρύματα:

1. Συνήθως, ένας φοιτητής παρακολουθεί μία συγκεκριμένη ομάδα μαθημάτων. Για παράδειγμα, τα μαθήματα σχηματίζουν ομάδες σύμφωνα με το εξάμηνο στο οποίο ανήκουν, ή τη γενική κατεύθυνση την οποία υπηρετούν· π.χ. μπορούμε να έχουμε τις ομάδες «Μαθήματα $1^{ου}$ Εξαμήνου», «Μαθήματα Μαθηματικών» κ.λπ. Δύο μαθήματα που ανήκουν στην ίδια ομάδα δεν θα πρέπει να γίνονται την ίδια χρονική στιγμή. Το καλύτερο που θα μπορούσε να γίνει είναι να προγραμματιστούν οι αντίστοιχες διαλέξεις (της κάθε ομάδας) να γίνονται όσο το δυνατόν πιο κοντά, έτσι ώστε να μην υπάρχουν κενά στα προσωπικά ωρολόγια προγράμματα των περισσοτέρων φοιτητών. Κάτι άλλο που συνήθως απαιτείται είναι η ισοκατανομή των διαλέξεων μιας ομάδας κατά τη διάρκεια μιας εβδομάδας.
2. Συνήθως, κάθε μάθημα χωρίζεται σε διαλέξεις με προκαθορισμένη διάρκεια. Έτσι, κάθε σύστημα κατάρτισης ωρολογίων προγραμμάτων θα πρέπει να λαμβάνει υπόψη αυτή την πληροφορία. Δύο διαλέξεις του ιδίου μαθήματος δεν μπορούν να γίνονται την ίδια ημέρα.

[2] Για απλότητα, θεωρούμε ότι το C είναι ένα σύνολο από θετικούς ακεραίους.

Τα στοιχεία των συνόλων που ορίσαμε παραπάνω, έχουν τις δικές τους ιδιότητες. Κάθε $c_i \in C$, $t_i \in T$ και $s_i \in S$ έχει αντίστοιχα την ιδιότητα

$$\begin{aligned} \mathrm{nac}(c_i) &\subseteq P, \\ \mathrm{nat}(t_i) &\subseteq P, \\ \mathrm{nas}(s_i) &\subseteq P. \end{aligned}$$

Το πρόθεμα «na-» προέρχεται από το «not available», συνεπώς αυτή η ιδιότητα αφορά τις διδακτικές περιόδους στις οποίες η οντότητα στην οποία αναφέρεται δεν είναι διαθέσιμη. Π.χ. το $\mathrm{nat}(t_2) = \{0, 5\}$ σημαίνει ότι ο δάσκαλος t_2 δεν είναι διαθέσιμος (για να διδάξει) τις περιόδους 0 και 5.

Κάθε μάθημα $s_i \in S$ συνδέεται με τις αίθουσες και τους διδάσκοντές του μέσω των συναρτήσεων

$$\begin{aligned} \emptyset \neq \mathrm{class}(s_i) &\subseteq C, \\ \emptyset \neq \mathrm{teach}(s_i) &\subseteq T. \end{aligned}$$

Η πρώτη είναι το σύνολο των αιθουσών που μπορούν να φιλοξενήσουν το s_i και η δεύτερη είναι το σύνολο των υπεύθυνων δασκάλων για αυτό.

$$\begin{aligned} \mathrm{subj} : &\ L \to S, \\ \mathrm{dur} : &\ L \to \{1, 2, \ldots, H\}. \end{aligned}$$

Η συνάρτηση «subj» απεικονίζει το L στο S, δηλαδή το $\mathrm{subj}(\ell)$ είναι το μάθημα στο οποίο ανήκει η διάλεξη ℓ. Η συνάρτηση «dur» καθορίζει τη διάρκεια κάθε διάλεξης. Τέλος, η συνάρτηση

$$\mathrm{imp} : G \to \mathbb{N}$$

καθορίζει τη βαρύτητα κάθε ομάδας μαθημάτων· δηλαδή, αν το $\mathrm{imp}(g)$ είναι ένας μεγάλος αριθμός, τότε η ομάδα g έχει μεγάλη βαρύτητα στο εκπαιδευτικό ίδρυμα.

Για κάθε διάλεξη $\ell_i \in L$, ζητούμε να βρούμε τον χρόνο και τον χώρο στον οποίον θα λάβει χώρα. Συμπερασματικά, μία λύση στο πρόβλημα είναι ο πλήρης ορισμός της συνάρτησης $\mathrm{sol} : L \to P \times C$, η οποία συνδέει κάθε διάλεξη με μία διδακτική περίοδο έναρξής της και μία αίθουσα.

3 Ορισμός του Προβλήματος Ικανοποίησης Περιορισμών

Ορίζουμε το πρόβλημα που περιγράφηκε παραπάνω σαν ένα *Πρόβλημα Ικανοποίησης Περιορισμών – ΠΙΠ* (Constraint Satisfaction Problem – CSP).

Ένα ΠΙΠ περιέχει ένα σύνολο από *περιορισμένες μεταβλητές* (constrained variables) — μπορούμε να τις αναφέρουμε απλά και σαν *μεταβλητές*· κάθε μεταβλητή αντιστοιχεί σε ένα *πεδίο τιμών* (domain). Λέμε ότι μία περιορισμένη μεταβλητή x έχει πεδίο τιμών D_x. Οι περιορισμένες μεταβλητές συνδέονται μεταξύ τους μέσα από ένα σύνολο *περιορισμών* (constraints). Γενικά, ένας περιορισμός που αφορά συγκεκριμένες περιορισμένες μεταβλητές είναι ένα σύνολο με όλους τους έγκυρους συνδυασμούς τιμών που μπορούν να ανατεθούν. Για παράδειγμα, για τις μεταβλητές

x_1 και x_2 με πεδία τιμών $\{0, 1, 2, 3\}$, ο περιορισμός της ισότητας μπορεί να δηλωθεί σαν $\mathcal{C}(\{x_1, x_2\}, \{(0, 0), (1, 1), (2, 2), (3, 3)\})$. Παρότι αυτός ο συμβολισμός είναι όσο πιο γενικός γίνεται, στην πράξη (δηλαδή στον Προγραμματισμό με Περιορισμούς) χρησιμοποιούμε απλές σχέσεις για να περιγράψουμε τα δίκτυα περιορισμών. Στο παραπάνω παράδειγμα, ο περιορισμός μπορεί να γραφεί απλά σαν $x_1 = x_2$. Στην παρούσα εργασία, θα χρησιμοποιηθούν μόνο οι κατηγορίες περιορισμών που είναι υλοποιημένες στους περισσότερους επιλυτές ΠΙΠ. Τέλος, μία *λύση* σε ένα ΠΙΠ είναι μία έγκυρη ανάθεση μίας τιμής σε κάθε μεταβλητή, η οποία ικανοποιεί όλους τους περιορισμούς.

3.1 Μεταβλητές και Πεδία Τιμών

Έχουμε το σύνολο των περιορισμένων μεταβλητών

$$\mathscr{X} = \{x_{i,k} \mid D_{x_{i,k}} = P \backslash \mathrm{nas}(\mathrm{subj}(\ell_i)),\ \ell_i \in L,\ 1 \leq k \leq \mathrm{dur}(\ell_i)\}$$

που αναφέρεται στις διδακτικές περιόδους κάθε διάλεξης. Με άλλα λόγια, για μια διάλεξη ℓ_i έχουμε τις περιορισμένες μεταβλητές $x_{i,1}, x_{i,2}, \ldots, x_{i,\mathrm{dur}(\ell_i)}$. Στην πραγματικότητα, η κύρια μεταβλητή είναι η $x_{i,1}$, αφού οι διδακτικές περίοδοι είναι διαδοχικές για κάθε διάλεξη και η διάρκειά της είναι γνωστή. Έτσι, για απλότητα, ορίζουμε $x_i \equiv x_{i,1}$. Το άλλο σημαντικό σύνολο περιορισμένων μεταβλητών

$$\mathscr{C} = \{cl_i \mid D_{cl_i} = \mathrm{class}(\mathrm{subj}(\ell_i)),\ \ell_i \in L\}$$

περιλαμβάνει τις αίθουσες όπου θα παραδίδονται οι διαλέξεις. Οι υπόλοιπες περιορισμένες μεταβλητές που θα χρησιμοποιήσουμε είναι βοηθητικές.

3.2 Περιορισμοί για τις Διαλέξεις

Έχοντας ορίσει όλες τις σημαντικές μεταβλητές και τα πεδία τιμών τους, θα «χτίσουμε» τώρα το δίκτυο περιορισμών. Ένας προφανής περιορισμός είναι ότι οι περίοδοι τις ίδιας διάλεξης πρέπει να είναι διαδοχικές:

$$x_{i,p+1} = x_{i,p} + 1, \quad 1 \leq p < \mathrm{dur}(\ell_i), \quad \forall \ell_i \in L. \tag{1}$$

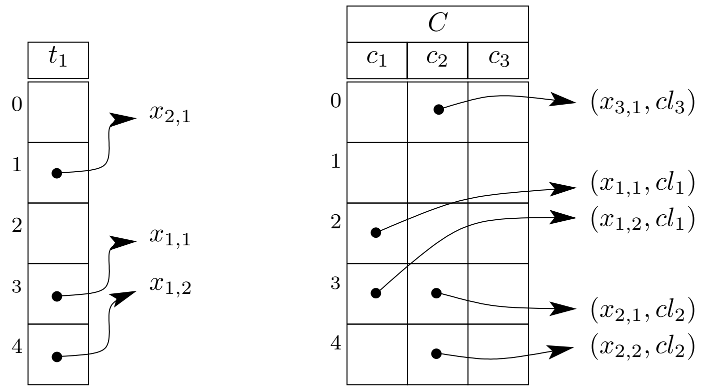
Για να εξασφαλίσουμε ότι οι διαλέξεις του ίδιου μαθήματος δεν παραδίδονται την ίδια ημέρα, για κάθε ℓ_i και ℓ_j με $i \neq j$ και $\mathrm{subj}(\ell_i) = \mathrm{subj}(\ell_j)$ θέτουμε τον περιορισμό:

$$\left\lfloor \frac{x_i}{H} \right\rfloor \neq \left\lfloor \frac{x_j}{H} \right\rfloor. \tag{2}$$

Φυσικά, μία διάλεξη δεν μπορεί να ξεκινά στο τέλος της μίας ημέρας και να τελειώνει στην αρχή της επόμενης, παρ' όλο που οι περιορισμοί που έχουμε δηλώσει έως τώρα το επιτρέπουν αυτό! (Για παράδειγμα, η ανάθεση $x_{i,1} \leftarrow H-1$, $x_{i,2} \leftarrow H$ είναι έγκυρη —με τους έως τώρα δηλωμένους περιορισμούς— για μια διάλεξη ℓ_i, με $\mathrm{dur}(\ell_i) \geq 2$, που προγραμματίζεται να παραδίδεται σε δύο διαφορετικές ημέρες!) Για να μην ξεκινούν λοιπόν οι διαλέξεις «πολύ αργά», μπορούμε αντί της προσθήκης ενός νέου περιορισμού, να μειώσουμε τα αρχικά πεδία τιμών των μεταβλητών $x_i \in \mathscr{X}$.

$$D_{x_i} \leftarrow D_{x_i} \backslash \bigcup_{d=0}^{D-1} \{(d+1)H - \mathrm{dur}(\ell_i) + 1, \ldots, (d+1)H - 1\}. \tag{3}$$

3.3 Πόροι και Περιορισμοί



(α′) Ένας δάσκαλος-πόρος t_1 για $|P| = 5$.

(β′) Ο πόρος για τις αίθουσες (σύνολο C) είναι δισδιάστατος.

Σχήμα 1. Πόροι και αναθέσεις τους.

Για να κατανοήσει κανείς την προσέγγισή μας στο πρόβλημα, θα πρέπει να δει τους δασκάλους, τις αίθουσες και τις ομάδες μαθημάτων σαν *πόρους*. Κάθε πόρος συνδέεται με ένα σύνολο δραστηριοτήτων. Ειδικότερα, για κάθε δάσκαλο $t \in T$ έχουμε $\mathscr{X}_t = \{x_{i,k} \mid x_{i,k} \in \mathscr{X},\ t \in \text{teach}(\text{subj}(\ell_i))\}$ και για κάθε ομάδα μαθημάτων $g \in G$, $\mathscr{X}_g = \{x_{i,k} \mid x_{i,k} \in \mathscr{X},\ \text{subj}(\ell_i) \in g\}$. Ο καθολικός περιορισμός που θα πρέπει να δηλωθεί για όλα τα παραπάνω σύνολα είναι το «να πάρουν οι μεταβλητές του κάθε συνόλου διαφορετικές μεταξύ τους τιμές» (περιορισμός «all-different»). (Κάθε πόρος μπορεί να παρέχεται σε μία το πολύ δραστηριότητα/διάλεξη κατά τη διάρκεια μία διδακτικής περιόδου. Για παράδειγμα, στο Σχήμα 1(α′), ο δάσκαλος t_1 παραδίδει μία δίωρη διάλεξη ℓ_1 και μία μονόωρη διάλεξη ℓ_2. Δεν μπορούμε να αναθέσουμε ίδιες τιμές στις αντίστοιχες περιορισμένες μεταβλητές. Ο πίνακας με τις «υποδοχές» μπορεί να ιδωθεί σαν το προσωπικό ωρολόγιο πρόγραμμα του t_1.)

$$\text{AllDiff}(\mathscr{X}_t), \quad \forall t \in T, \tag{4}$$
$$\text{AllDiff}(\mathscr{X}_g), \quad \forall g \in G. \tag{5}$$

Δεν θα πρέπει να λησμονήσουμε να μειώσουμε τα αρχικά πεδία τιμών των μεταβλητών στο $\mathscr{X}_t$ σύμφωνα με το

$$D_{x_{i,k}} \leftarrow D_{x_{i,k}} \backslash \text{nat}(t), \quad \forall x_{i,k} \in \mathscr{X}_t,\ \forall t \in T. \tag{6}$$

Από την άλλη πλευρά, έχουμε ένα μόνο σύνολο με τις περιορισμένες μεταβλητές ολόκληρου του συνόλου των αιθουσών C. Ορίζουμε $\mathscr{X}_C = \{y_{i,k} \mid y_{i,k} = x_{i,k} + D \cdot H \cdot cl_i,\ x_{i,k} \in \mathscr{X},\ cl_i \in \mathscr{C}\}$. Κάθε μέλος του $\mathscr{X}_C$ αντιπροσωπεύει τον χρόνο *και* τον χώρο όπου μία διάλεξη λαμβάνει χώρα. Η έκφραση $x_{i,k} + D \cdot H \cdot cl_i$ είναι η γραμμική αναπαράσταση του $(x_{i,k}, cl_i)$, αφού το $x_{i,k}$ δεν μπορεί να υπερβεί το $D \cdot H - 1$. (Κάναμε αυτό το «τρικ» γραμμικοποίησης, για να αποφύγουμε τη δήλωση δισδιάστατων περιορισμένων μεταβλητών, που είναι πιο δύσκολες στον χειρισμό.) Έτσι, προσθέτουμε έναν ακόμα περιορισμό

$$\mathrm{AllDiff}(\mathscr{X}_C). \tag{7}$$

Για την καλύτερη κατανόηση του παραπάνω περιορισμού, ας δούμε το Σχήμα 1(β′). Εκεί έχουμε μία δισδιάστατη απεικόνιση του πόρου «αίθουσες», γιατί σε μία διάλεξη μπορούμε να αναθέσουμε μία περίοδο από το P και μία αίθουσα από το C. Στο σχήμα, έχουμε αναθέσεις χρόνου και χώρου για τις διαλέξεις ℓ_1, ℓ_2 και ℓ_3, με αντίστοιχες διάρκειες 2, 2 και 1· επίσης, $D \cdot H = |P| = 5$ και $|C| = 3$. Οποιεσδήποτε δύο διαλέξεις δεν μπορούν να μοιράζονται την ίδια περίοδο *και* αίθουσα.

Για ακόμα μια φορά, τα αρχικά πεδίων τιμών των μεταβλητών στο $\mathscr{X}_C$ οφείλουν να είναι συμβατά με τις δηλωμένες διαθεσιμότητες των αιθουσών:

$$D_y \leftarrow D_y \backslash \{y \mid y = p + D \cdot H \cdot c_i,\ p \in \mathrm{nac}(c_i),\ c_i \in C\}, \quad \forall y \in \mathscr{X}_C. \tag{8}$$

3.4 Αντικειμενική Μεταβλητή

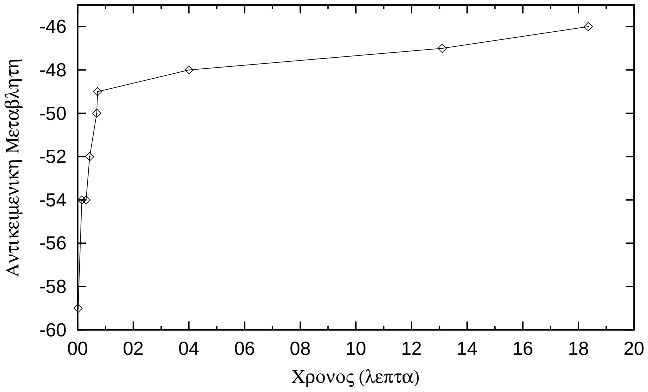
Σε πολλούς επιλυτές, μία περιορισμένη μεταβλητή χρησιμοποιείται για την περιγραφή της αντικειμενικής συνάρτησης (objective function) ενός προβλήματος. Η *αντικειμενική μεταβλητή* όπως την ονομάζουμε, αντιπροσωπεύει το κόστος, ή, καλύτερα, την ποιότητα του ωρολογίου προγράμματος. Δεν είναι τετριμμένο να ορίσουμε μία τέτοια μετρική. Η προσέγγισή μας έγκειται στον συνδυασμό δύο κριτηρίων ποιότητας.

Ισοκατανομή των διαλέξεων μέσα στην εβδομάδα Ο πρώτος στόχος είναι η ομαλή κατανομή των διαλέξεων κάθε ομάδας μαθημάτων $g \in G$. Με άλλα λόγια προσπαθούμε να ελαχιστοποιήσουμε την παράσταση

$$\sum_{g \in G} \mathrm{imp}(g) \left(\max_{0 \le d < D} \{\mathsf{HoursPerDay}_g[d]\} - \min_{0 \le d < D} \{\mathsf{HoursPerDay}_g[d]\} \right). \tag{9}$$

Παρατηρούμε ότι οι όροι του αθροίσματος έχουν βάρη $\mathrm{imp}(g)$ τα οποία πολλαπλασιάζονται με τη διαφορά ανάμεσα στον αριθμό των περιόδων της ημέρας με τις περισσότερες διαλέξεις του g και της ημέρας με τις λιγότερες διαλέξεις του g. Ο αριθμός των περιόδων για τις διαλέξεις του g την ημέρα d συμβολίζεται με $\mathsf{HoursPerDay}_g[d]$. Πρόκειται για μία περιορισμένη μεταβλητή, η οποία είναι δυνατόν να παραχθεί μέσω απλών περιορισμών, από τις ήδη δηλωθείσες μεταβλητές.

Κενές περίοδοι μεταξύ διαλέξεων Κατά τον ίδιο τρόπο, γίνεται να ορίσουμε μία ακόμα μετρική που θα δίνει τα κενά μεταξύ των διαλέξεων που ανήκουν στην ίδια ομάδα μαθημάτων. Εφόσον έχουμε λιγότερα κενά, οι κενές ώρες για τους μαθητές θα είναι λίγες και έτσι το ωρολόγιο πρόγραμμα θα είναι καλύτερο.



Σχήμα 2. Πορεία βελτίωσης των λύσεων για ένα πραγματικό πρόβλημα.

4 Αναζήτηση Λύσεων

Αφού έχουμε ορίσει το ΠΙΠ, πρέπει τώρα να διαλέξουμε μία μέθοδο για να το επιλύσουμε. Γενικά, μπορούμε να αναθέτουμε τιμές στις μεταβλητές του $\mathscr{X}$ και του $\mathscr{C}$ και να ελέγχουμε αν ικανοποιούν τους περιορισμούς. Όμως υπάρχουν πολλοί τρόποι για να το κάνουμε αυτό.

Έχει δειχθεί ότι για τα περισσότερα προβλήματα η *Διατήρηση Συνέπειας Ακμών* (Maintaining Arc Consistency – MAC) δίνει καλά αποτελέσματα [15,16]. Αυτή η μεθοδολογία υποδεικνύει ότι όποτε αναθέτουμε μία τιμή σε μία περιορισμένη μεταβλητή (ή, γενικότερα, όταν περιορίζουμε το πεδίο τιμών της), είναι καλό να εφαρμόζουμε συνέπεια-ακμών στο δίκτυο περιορισμών, με σκοπό να «κλαδέψουμε» τα τμήματα εκείνα του δένδρου αναζήτησης στα οποία δεν βρίσκεται καμία λύση.

Ορίζουμε ότι μία ακμή (x, x') —που συνδέει μεταβλητές του δικτύου περιορισμών— είναι συνεπής, αν για όλες τις τιμές $v \in D_x$, υπάρχει μία τιμή $v' \in D_{x'}$ τέτοια ώστε να ικανοποιείται ο περιορισμός που συνδέει τις δύο μεταβλητές. Όταν κάθε ακμή του δικτύου περιορισμών είναι συνεπής, λέμε ότι το δίκτυο περιορισμών χαρακτηρίζεται από *συνέπεια-ακμών* (arc-consistency). Η συνέπεια-ακμών δεν οδηγεί απαραίτητα σε μία λύση —αλλά αν δεν υπάρχει συνέπεια-ακμών, είμαστε σίγουροι ότι δεν υπάρχει λύση—, εκτός εάν τη συνδυάσουμε με μία μέθοδο αναζήτησης. Η χρησιμότητά της έχει να κάνει με τη μείωση του χώρου αναζήτησης τον οποίο μία συνηθισμένη μέθοδος αναζήτησης με την οποία συνδυάζεται —όπως η πρώτα-κατά-βάθος αναζήτηση (Depth First Search – DFS), η αναζήτηση περιορισμένων ασυμφωνιών (Limited Discrepancy Search – LDS) κ.ά.— πρέπει να εξερευνήσει. Σε αυτήν την εργασία, χρησιμοποιούμε κυρίως τη Βελτιωμένη μέθοδο LDS [11], στο πνεύμα πάντα της Διατήρησης Συνέπειας Ακμών.

Εφόσον γνωρίζουμε μία λύση σε ένα ΠΙΠ, μπορούμε να χρησιμοποιήσουμε στο τέλος Τοπική Αναζήτηση (Local Search), σαν μία τελευταία προσπάθεια βελτίωσης της αντικειμενικής μεταβλητής. Μία συνηθισμένη πρακτική σε αυτό το πλαίσιο είναι το επαναλαμβανόμενο «πάγωμα» των μεταβλητών στις τιμές που πήραν στην τελευταία λύση που έχει βρεθεί· από το «πάγωμα» αυτό εξαιρούμε κάθε φορά μία συγκεκριμένη γειτονιά μεταβλητών (δηλαδή ένα υποσύνολο των μεταβλητών με διάφορα κοινά χαρακτηριστικά), στις οποίες δοκιμάζουμε αναθέσεις τιμών έτσι ώστε να φτάσουμε σε καλύτερη λύση.

Το Σχήμα 2 συνοψίζει την όλη διαδικασία της αναζήτησης και παρουσιάζει τις λύσεις που έχουν βρεθεί για ένα πραγματικό πρόβλημα (το πρόβλημα κατάρτισης ωρολογίου προγράμματος για το Τμήμα Πληροφορικής και Τηλεπικοινωνιών) και πώς αυτές βελτιώνονται. Αυτό το πρόβλημα περιλαμβάνει 40 καθηγητές, 51 μαθήματα, 13 ομάδες μαθημάτων και 8 αίθουσες. Μετά το 15^{o} λεπτό, η αναζήτηση γίνεται μεταευριστική· συγκεκριμένα, από αυτό το σημείο και μετά εφαρμόζουμε Τοπική Αναζήτηση στην τελευταία λύση και βλέπουμε ότι η λύση βελτιώνεται.

5 Συμπεράσματα και Μελλοντική Δουλειά

Ο Προγραμματισμός με Περιορισμούς είναι ένας δοκιμασμένος τρόπος επίλυσης προβλημάτων βελτιστοποίησης. Παρουσιάστηκε μία αποτελεσματική εφαρμογή του στο πρόβλημα κατάρτισης ωρολογίων προγραμμάτων. Η μοντελοποίηση του προβλήματος σαν ένα ΠΙΠ μάς επέτρεψε να εκμεταλλευτούμε αρκετές γενικές μεθόδους αναζήτησης, αφού η διατύπωση του προβλήματος και η επίλυση του είναι ανεξάρτητες φάσεις στον Προγραμματισμό με Περιορισμούς. Οπότε, μπορεί κανείς να πειραματιστεί και με άλλες, παλιές και μοντέρνες μεθόδους αναζήτησης και να τις συνδυάσει· η δήλωση του προβλήματος θα παραμείνει ως έχει, εκτός αν κάποιος θέλει να προσθέσει καινούργιους περιορισμούς ή να γενικεύσει ακόμα περισσότερο το πρόβλημα.

Τέλος, υπάρχουν πολλές δουλειές σχετικές με την επιβολή συνέπειας ακμών από έναν επιλυτή πάνω σε δίκτυα περιορισμών. Στο μέλλον, οι επιλυτές ΠΙΠ αναμένεται να είναι ταχύτεροι και έτσι η αναζήτηση θα γίνει αποτελεσματικότερη. Θα ήταν λοιπόν πολύ σημαντικό να υπάρξει μία προσπάθεια ενοποίησης των περιβαλλόντων Προγραμματισμού με Περιορισμούς, έτσι ώστε όλες οι μοντελοποιήσεις ΠΙΠ —σαν και αυτήν που παρουσιάσαμε— να υποστηρίζονται άμεσα από πληθώρα επιλυτών.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον συνεργάτη του Τμήματος, Κυριάκο Ζερβουδάκη, για τις υποδείξεις του σχετικά με το πώς μπορεί να διατυπωθεί αντικειμενοστραφώς και να επιλυθεί ένα πρόβλημα κατάρτισης ωρολογίου προγράμματος.

Αναφορές

1. V. A. Bardadym. Computer-aided school and university timetabling: The new wave. In *Proc. 1st Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT'95)*, pages 22–45. Springer, 1995.
2. S. C. Brailsford, C. N. Potts, and B. M. Smith. Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research*, 119(3):557–581, 1999.

3. E. K. Burke, B. MacCarthy, S. Petrovic, and R. Qu. Multiple-retrieval case-based reasoning for course timetabling problems. *Journal of the Operational Research Society (JORS)*, 57(2):148–162, 2005.
4. E. K. Burke and S. Petrovic. Recent research directions in automated timetabling. *European Journal Of Operational Research*, 140(2):266–280, 2002.
5. D. Costa. A tabu search algorithm for computing an operational timetable. *European Journal of Operational Research*, 76(1):98–110, 1994.
6. D. de Werra. Introduction to timetabling. *European Journal of Operational Research*, 19:151–162, 1985.
7. S. Elmohamed, P. Coddington, and G. Fox. A comparison of annealing techniques for academic course scheduling. In *Proc. 2^{nd} Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT'97)*, volume 1408, pages 92–112. Springer, 1997.
8. J. A. Ferland. Generalized assignment-type problems: A powerful modeling scheme. In *Proc. 2^{nd} Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT'97)*, pages 53–77. Springer, 1997.
9. H. Frangouli, V. Harmandas, and P. Stamatopoulos. UTSE: Construction of optimum timetables for university courses – A CLP based approach. In *Proc. 3^{rd} Int. Conf. on the Practical Applications of Prolog (PAP'95)*, pages 225–243, 1995.
10. Y. Kochetov, P. Obuhovskaya, and M. Paschenko. Local search heuristics for the teacher/class timetabling problem. *Proc. 6^{th} Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT'06)*, pages 454–457, 2006.
11. Richard E. Korf. Improved limited discrepancy search. In *13^{th} National Conf. on Artificial Intelligence (AAAI'96)*, pages 286–291, 1996.
12. P. Kostuch. The university course timetabling problem with a 3-phase approach. *Proc. 5^{th} Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT'04)*, pages 251–266, 2004.
13. P. Mellor. A review of job shop scheduling. *OR*, 17(2):161–171, 1966.
14. R. Ostermann and D. de Werra. Some experiments with a timetabling system. *OR Spectrum*, 3(4):199–204, 1983.
15. D. Sabin and E. C. Freuder. Contradicting conventional wisdom in constraint satisfaction. In *Proc. 2^{nd} Int. Workshop on Principles and Practice of Constraint Programming (PPCP'94)*, pages 125–129, 1994.
16. D. Sabin and E. C. Freuder. Understanding and improving the MAC algorithm. In *Proc. 3^{rd} Int. Conf. on Principles and Practices of Constraint Programming (CP'97)*, pages 167–181, 1997.
17. A. Schaerf and L. Di Gaspero. Measurability and reproducibility in timetabling research: State-of-the-art and discussion. *Proc. 6^{th} Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT'06)*, pages 53–62, 2006.
18. P. Stamatopoulos, E. Viglas, and S. Karaboyas. Nearly optimum timetable construction through CLP and intelligent search. *Int. Journal on Artificial Intelligence Tools*, 7(4):415–442, 1998.
19. R. Weare, E. Burke, and D. Elliman. A hybrid genetic algorithm for highly constrained timetabling problems. In *Proc. 6^{th} Int. Conf. on Genetic Algorithms*, pages 605–610. LJ Eshelman, 1995.
20. G. M. White and P. W. Chan. Towards the construction of optimal examination timetables. *INFOR*, 17:219–229, 1979.
21. K. Zervoudakis and P. Stamatopoulos. A generic object-oriented constraint based model for university course timetabling. In *Proc. 3^{rd} Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT'00)*, pages 128–147. Springer, 2000.